

### AMENDMENTS TO THE CLAIMS

1. (Currently Amended) An analysis method, comprising:  
identifying at least one construct in a program, wherein the program ~~is configured for~~ comprises native instructions for execution on a first processor having a first machine instruction set;  
assigning at least one native instruction of the program proximate the construct to be a trigger to invoke analysis code; and  
interpreting, via an interpreter, the program on a second processor[,]  
having a second machine instruction set; ~~wherein~~  
during the interpretation[,]  
and upon interpreting the assigned native instruction,  
identifying the assigned native instruction as the trigger; and  
invoking the analysis code is invoked by the interpreter at the identified construct[s] in response to identifying the trigger wherein the analysis code includes machine instructions of the second machine instruction set for execution directly on the second processor, and wherein the analysis code and the interpreter communicate via a predefined interface.
2. (Currently Amended) The method of Claim 1, wherein the construct is selected from the group comprising: a basic block of native instructions, ~~an~~ a specified native instruction, a group of native instructions, and a native procedure.
3. (Currently Amended) The method of Claim 1, wherein during the identification of a construct, ~~a~~ the trigger is created and information about the trigger is stored in a file for usage by the interpreter.
4. (Currently Amended) The method of Claim 1, wherein during the identification of a construct, [a] the trigger is created and information about the trigger is stored in the program.
5. (Original) The method of Claim 1, wherein the predefined interface includes a registration procedure for the analysis code to register with the interpreter.
6. (Original) The method of Claim 1, wherein invoking the analysis code consists of providing to the analysis code at least one item selected from the group comprising: a null statement, a register value, a memory value, a program counter address, branch instructions, and an effective address.

7. (Original) The method of Claim 1, additionally comprising registering the analysis code with the interpreter via a predefined application programming interface.

8. (Original) The method of Claim 1, wherein interpretation comprises emulation.

9. (Original) The method of Claim 1, wherein interpretation comprises simulation.

10. (Currently Amended) A method, comprising:

storing a compiled analysis binary program, wherein the analysis binary program includes machine instructions from a first machine instruction set, and wherein the analysis binary program is configured to analyze or trace state information of an interpretable program, wherein the interpretable program comprise native machine instructions of a second machine instruction set;

assigning at least one native instruction of the interpretable program proximate a selected construct to be a trigger to invoke the analysis binary program; and

interpreting, via an interpreter comprising machine instructions from the first machine instruction set and executing on a processor configured to execute the first machine instruction set, the interpretable program ~~for execution on a processor, wherein the interpretable program includes machine instructions from a second machine instruction set,~~

~~wherein the processor is configured to execute machine instructions from the first machine instruction set, and~~

~~wherein during the interpreting, upon encountering the assigned native instruction proximal~~ a selected construct in the interpretable program, identifying the assigned native instruct as the trigger and invoking the analysis binary program ~~is invoked by the interpreter and is provided~~ providing at least one item of state information about the execution of the interpretable program in response to identifying the trigger.

11. (Original) The method of Claim 10, wherein the state information includes register values, parameter values, instruction addresses, or data addresses.

12. (Original) The method of Claim 10, wherein the second machine instruction set includes generic machine instructions that are configured to be emulated on heterogeneous hardware platforms.

13. (Original) The method of Claim 10, wherein the construct comprises a procedure.

14. (Original) The method of Claim 10, wherein the construct comprises an instruction.

15. (Original) The method of Claim 10, wherein the interpretable program is a binary program configured for direct execution on a second processor.

16. (Previously Presented) The method of Claim 10, additionally comprising:  
storing location information about the selected construct in a file; and  
using the file during the interpretation so as to identify the selected construct.

17. (Currently amended) The method of Claim 10, additionally comprising inserting a the native trigger instruction proximate to the selected construct, ~~and wherein an interpreter is configured to recognize the trigger instruction as an instruction to invoke the analysis binary program.~~

18. (Canceled).

19. (Original) The method of 17, wherein the inserted trigger instruction is a machine instruction that does not substantially affect the performance of the interpretable program.

20. (Original) The method of 17, wherein the inserted trigger instruction is a no-op machine instruction.

21. (Original) The method of Claim 10, wherein interpretation comprises emulation.

22. (Original) The method of Claim 10, wherein interpretation comprises simulation.

23. (Original) The method of Claim 10, additionally comprising ignoring selected machine interactions in the interpretable program.

24. (Currently Amended) An analysis or testing system, comprising:

a storage means for storing an analysis binary code, wherein the analysis binary code includes machine instructions from a first machine instruction set, and wherein the binary code is configured to analyze or trace state information of an interpretable program; and

a processor configured to execute an interpreter program ~~means~~ for interpreting the interpretable program ~~for execution on a processor~~, wherein the interpretable program includes machine instructions from a second machine instruction set,

wherein the processor is configured to execute machine instructions from the first machine instruction set, and

wherein the processor is configured to receive an assignment of at least one native instruction proximate a selected construct of the interpretable program to be a trigger to invoke the analysis binary code and during the interpreting, upon encountering and interpreting the assigned native instruction, a selected construct in the program, the processor is configured to identify the assigned native instruction as the trigger and a selected condition is processed and the binary code is conditionally invoke[d] the binary code and is provide[d] at least one item of state information about the execution of the program in response to identifying the trigger.

25. (Original) The system of Claim 24, wherein the second machine instruction set includes generic machine instructions that configured to be capable of being emulated on heterogeneous hardware platforms.

26. (Currently Amended) The system of Claim 24, additionally comprising:  
a second storage configured to means-for storing store information about the selected construct in a file; and

wherein the processor is further configured to means-for using use the file during the interpretation of the interpretable program so as to identify the selected construct.

27. (Currently Amended) The system of Claim 24, ~~additionally comprising whrein the~~ processor is further configured to inserting a the trigger instruction proximate to the selected construct, ~~and wherein an interpreter is configured to recognize the trigger instruction as an instruction to invoke the binary code.~~

28. (Canceled).

29. (Currently Amended) The ~~method~~ system of Claim 27, wherein interpretation comprises emulation.

30. (Currently Amended) The ~~method~~ system of Claim 27, wherein interpretation comprises simulation.

31. (Currently Amended) An analysis method, comprising:

assigning at least one native instruction of a first machine instruction set proximate a selected construct of a binary program to be a trigger to invoke analysis code;

interpreting [a] the binary program by an interpreter on a first processor, wherein the binary program is configured for native execution on a second processor,

wherein during the interpretation[,] and upon interpreting the assigned native instruction, identifying the assigned native instruction as the trigger; and

invoking, in response to the identifying, the analysis code is invoked by an the interpreter at a the trigger instruction in the binary program, wherein the analysis code includes machine instructions for processing directly on the first processor, and wherein the analysis code has been compiled prior to the execution of the interpreter; and

executing the binary program, including the trigger instruction, natively on the second processor.

32. (Original) The method of Claim 31, wherein invoking the analysis code comprises providing to the analysis code at least one item selected from the group comprising: a register value, a memory value, a program counter address, branch instructions, and an effective address.

33. (Original) The method of Claim 31, additionally comprising registering the analysis code with the interpreter via a predefined application programming interface.

34. (Original) The method of Claim 33, additionally comprising identifying at least one trigger using the predefined application programming interface, wherein encountering the trigger during interpretation causes the analysis code to be invoked.

35. (Original) The method of Claim 31, wherein interpretation comprises emulation.

36. (Original) The method of Claim 31, wherein interpretation comprises simulation.

37. (Currently Amended) A computer readable medium ~~comprising~~ having stored thereon instructions that when executed cause a first processor of computer to[,]:

assign at least one native instruction proximate a construct of a binary program to be a trigger to invoke analysis code; and

interpret by an interpreter a binary program on a the first processor, wherein the binary program includes machine instructions from a machine instruction set of a second processor, wherein during the interpretation, and upon encountering a selected machine instruction from the machine instruction set of the second processor:

identify the assigned native instruction as the trigger, and

invoke the analysis code is invoked by an the interpreter via a predefined interface and executed natively on the first processor in response to identifying the trigger, wherein

the analysis code includes native machine instructions for processing directly on the first processor.

38. (Previously Presented) The computer readable medium of Claim 37, additionally comprising a predefined application programming interface that is defined by the interpreter so as to allow the analysis code to register and to define one or more callback routines.

39. (Previously Presented) The computer readable medium of Claim 37, wherein interpretation comprises emulation.

40. (Previously Presented) The computer readable medium of Claim 37, wherein interpretation comprises simulation.

41. – 47. (Canceled).

48. The method of Claim 1, wherein the native trigger instruction comprises at least one machine instruction that does not substantially affect the performance of the program.

49. The method of Claim 48, wherein the native trigger instruction comprises at least one no-op instruction of the first machine instruction set.

50. The system of Claim 24, wherein the native trigger instruction comprises at least one machine instruction that does not substantially affect the performance of the interpretable

51. The system of Claim 50, wherein the native trigger instruction comprises at least one no-op instruction of the first machine instruction set.

52. The method of Claim 31, wherein the native trigger instruction comprises at least one machine instruction that does not substantially affect the performance of the binary program.

53. The method of Claim 52, wherein the native trigger instruction comprises at least one no-op instruction of the first machine instruction set.

54. The medium of Claim 37, wherein the selected machine instruction comprises at least one machine instruction that does not substantially affect the performance of the binary program.

55. The medium of Claim 54, wherein the selected machine instruction comprises at least one no-op instruction of the first machine instruction set.